(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **08328840 A**

(43) Date of publication of application: **13.12.96**

(51) Int. Cl **G06F 9/06**
G06F 9/44

(21) Application number: **07133184**

(22) Date of filing: **31.05.95**

(71) Applicant: **HITACHI LTD**

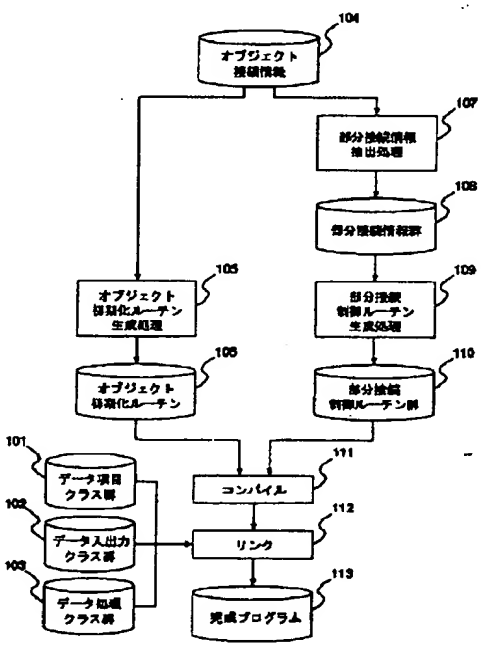
(72) Inventor: **CHIBA TOSHIYA**

(54) PROGRAM SYNTHESIZING METHOD

(57) Abstract:

PURPOSE: To synthesize a program by combining plural program components arbitrarily in the method to combine the plural program components and to automatically synthesize an application program.

CONSTITUTION: An object initialization routine 106 is generated and initialized in which an instance is generated from the combination of a data input/output object 102 and a data processing object 103 and object connection information 104 to define the transfer relation of data. Also, the connection information 104 is divided into plural partial connection information groups 108, and a partial connection control routine 110 to control an object at every partial connection is generated. An accomplished program 112 is obtained by compiling and linking generated routines 106, 110, a data item class group 101, a data input/output class group 102 and a data processing class group 103.



(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号

特開平8－328840

(43)公開日　平成８年(1996)12月13日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ	9/06	５３０		Ｇ０６Ｆ	9/06	５３０Ｖ
						５３０Ｗ
	9/44	５３０	7737－5B		9/44	５３０Ｐ

審査請求　未請求　請求項の数９　ＯＬ　（全 15 頁）

(21)出願番号　特願平7－133184

(22)出願日　平成７年(1995)５月31日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　千葉　俊哉
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(74)代理人　弁理士　小川　勝男

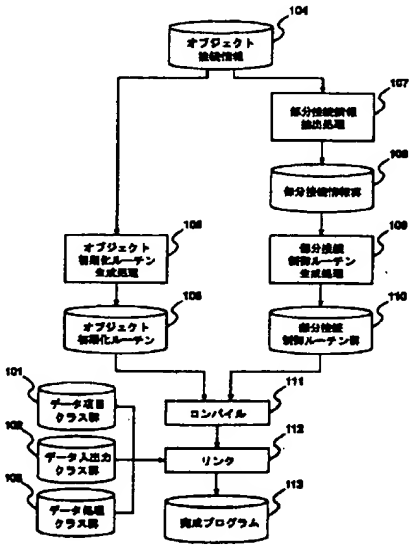
(54)【発明の名称】　プログラム合成方法

(57)【要約】

【目的】　複数のプログラム部品を組み合わせ、計算機によって自動的に応用プログラムを合成する方法において、複数のプログラム部品を任意に組み合わせてプログラムを合成することを可能とする。

【構成】　データ入出力オブジェクト102と、データ処理オブジェクト103の組み合わせと、データの受け渡し関係を定義するオブジェクト接続情報104から、インスタンスを生成・初期化するオブジェクト初期化ルーチン106を生成する。また、接続情報104を複数の部分接続情報群108に分解し、各部分接続毎にオブジェクトの制御を行う部分接続制御ルーチン群110を生成する。生成されたルーチン106、110と、データ項目クラス群101、データ入出力クラス群102、データ処理クラス群103をコンパイル・リンクして、完成プログラム112を得る。

図1



1

【特許請求の範囲】

【請求項１】計算機によって、仕様書の情報を元に、データ処理を行う応用プログラムを自動的に合成する際に、(1)応用プログラムで使用されるデータの、データ値を属性として持ち、該属性に対する操作をメソッドとして持つデータ項目オブジェクト、(2)データ入出力装置と、データ項目オブジェクトとの間でデータの受け渡しを行うデータ入出力オブジェクト、及び(3)他のオブジェクトから取得したデータ項目オブジェクトのメソッドを定型的な順序で呼び出す操作を行ない、前記データ項目オブジェクトを他のオブジェクトに渡すデータ処理オブジェクトであって、データ項目オブジェクトを取得する対象となるオブジェクト及び、データ項目オブジェクトを渡す対象となるオブジェクトがデータ入出力オブジェクトまたはデータ処理オブジェクトであるデータ処理オブジェクトを組み合わせて応用プログラムを合成するプログラム合成方法において、

応用プログラムにおいて使用される、データ入出力オブジェクトおよびデータ処理オブジェクトとの間のデータ項目オブジェクトの受け渡し関係を表現する、オブジェクト接続情報を入力し、該接続情報に記述された全てのデータ入出力オブジェクトとデータ処理オブジェクトの生成・初期化手続きを呼び出すコードを合成し、さらに該接続情報のうち、データ入出力オブジェクトのみからなる部分接続情報毎にループが存在する制御構造を持つコードを合成することを特徴とする、プログラム合成方法。

【請求項２】データ取得メソッドを持ち、該メソッドが呼び出されると、１ないし複数のデータ入出力オブジェクトおよびデータ処理オブジェクトのデータ取得メソッドを呼び出し、得られたデータ項目オブジェクトのメソッドを定型的な順序で呼び出した結果得られたデータ項目オブジェクトをメソッド呼出し元に返し、該手続きにおいて、データ取得メソッドを呼び出す対象となったオブジェクトがデータ処理オブジェクトである場合、該手続きと同様の手続きが再帰的に行われるように構成されている事を特徴とする、請求項１記載のプログラム合成方法。

【請求項３】データ項目オブジェクトをパラメータとして持つデータ書き込みメソッドを持ち、該メソッドが呼び出されると、パラメータとして渡されたデータ項目オブジェクトのメソッドを定型的な順序で呼び出した結果得られたデータ項目オブジェクトを引数として１ないし複数のデータ入出力オブジェクトおよびデータ処理オブジェクトの書き込みメソッドを呼び出し、該手続きにおいて、データ書き込みメソッドを呼び出す対象となったオブジェクトがデータ処理オブジェクトである場合、該手続きと同様の手続きが再帰的に行われるように構成されている事を特徴とする、請求項１記載のプログラム合成方法。

2

【請求項４】オブジェクト接続情報の、ノードがデータ処理オブジェクトのみで構成される部分接続情報について、同一リソースを使用するレコード入出力オブジェクトが複数接続される組み合わせを不正と判定する、オブジェクト接続情報検査処理を行うことを特徴とする、請求項１記載のプログラム合成方法。

【請求項５】上記部分接続情報において、複数のデータ処理オブジェクトが、単一のデータ処理オブジェクトからデータ項目オブジェクトを入力する接続を不正と判定する、オブジェクト接続情報検査処理を行うことを特徴とする、請求項１記載のプログラム合成方法。

【請求項６】上記部分接続情報において、複数のデータ処理オブジェクトが、単一のデータ処理オブジェクトにデータ項目オブジェクトを出力する接続を不正と判定する、オブジェクト接続情報検査処理を行うことを特徴とする、請求項１記載のプログラム合成方法。

【請求項７】上記部分接続情報において、ループ接続が存在する組み合わせを不正と判定する、オブジェクト接続情報検査処理を行うことを特徴とする、請求項１記載のプログラム合成方法。

【請求項８】オブジェクト接続情報を対話的に入力する手段を持ち、該入力手段を実行する際に、ユーザがレコード入出力構造体およびレコード処理構造体間のデータ項目オブジェクトの受け渡し関係を定義する都度、該定義が正当であるか否か判定することを特徴とする、請求項１記載のプログラム合成方法。

【請求項９】レコードの各フィールドで使用するデータ項目オブジェクトの種別を定義する事によって、データ項目オブジェクトの集合体であるレコードをデータ項目クラスとして生成し、データ入出力部品に格納されるデータ項目オブジェクトの種別を定義することによって、データ入出力部品を生成し、生成された該部品群を利用してプログラム合成を行う事を特徴とする、請求項１記載のプログラム合成方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は計算機を用いたプログラム自動合成ツールに関し、特に、バッチ処理プログラムのように、ファイル、端末等のデータ入力装置から順次データを入力し、入力されたデータに対して処理を行い、結果として得られたデータをプリンタ、端末等のデータ出力装置に順次出力するオブジェクト指向プログラムを計算機によって自動的に合成する、プログラム合成方法に関する。

【０００２】

【従来の技術】従来、プログラム合成方法としては、応用プログラムの構成要素である処理パターンを実行するプログラムを複数生成し、これらプログラム群を組み合わせ、順次実行する事によって目的とする動作を得る、特開平5-108319が知られていた。上記従来技術では、２

次記憶に作成されるテンポラリファイルを介し、プログラム間のデータ受け渡しを行っていた。

【０００３】

【発明が解決しようとする課題】上記従来方法では、運用時に管理しなければならないプログラムの本数がふえてしまい、運用に手間がかかるという問題があった。また、プログラム間でのデータ受け渡しは、２次記憶に作成されるテンポラリファイルを介して行われるため、２次記憶を無駄に消費する上に、プログラム実行性能が悪くなるという問題があった。

【０００４】上記の問題を解決するためには、処理パターンをプログラム部品として実現し、複数のプログラム部品を組み合わせて１本のプログラムに合成する手段を有し、かつ、テンポラリファイルを介さずに、プログラム部品間でデータ受け渡しを可能にしなければならない。上記事項を実現するためには、解決しなければならない３つの課題がある。

【０００５】第１の課題は、プログラム部品間で受け渡しされるデータの形式に関係なく、プログラム部品を自由に組み合わせられることである。プログラム部品間のインターフェースが、受け渡しされるデータの形式に依存していると、プログラム部品の組合せに複雑な制限が生じてしまう。

【０００６】第２の課題は、複数プログラム部品の動作シーケンスを取り扱う手段を有する事である。第１のプログラム部品で何らかのデータを処理し、その結果として得られたデータを第２のプログラム部品が処理する場合について考察する。第１のプログラム部品で入力データを全て処理した後に、第２のプログラム部品が処理を行うシーケンスとすると、処理途中のデータを保持する領域が必要となる。多量のデータを処理する場合、データを保持する領域が大きくなり、現実のシステムに適用することができない。従って、入力となるデータを一定の単位(レコード等)に区切り、単位毎に第１のプログラム部品と第２のプログラム部品を続けて動作させるような動作シーケンスにする必要がある。プログラム部品の動作シーケンスは複雑であり、合成されたプログラムの保守が困難になる。

【０００７】第３の課題は、プログラム部品の組み合わせの誤りを容易に発見する手段を、ユーザに提供する事である。複数のプログラム部品をつなぎあわせると、例えば複数のプログラム部品が同一のファイルに、同時に書き込み動作を行うなどの不正な動作を行う場合がある。第２の課題で述べたように、プログラム部品の動作シーケンスを扱う問題は複雑であり、ユーザがこのような問題を考慮し、プログラム部品の組み合わせの誤りを発見することは容易ではない。

【０００８】本発明は、上記３つの問題を解決することにより、複数のプログラム部品をつなぎ合わせて1本のプログラムを合成することを可能とし、より実用性の高いプログラム合成方法を提供する事を目的とする。

【０００９】

【課題を解決するための手段】本発明のプログラム合成方法では、オブジェクト指向プログラミング技術を用い、大きく分けて次の３種類のオブジェクトを組み合わせることによってプログラムを合成する。

【００１０】(1)データ項目オブジェクト

特定の種別のデータを抽象化したオブジェクト。データ種別により異なる種類のデータ項目オブジェクトが存在する。データ値を属性として持ち、データ値に対する基本的な操作(大小比較や文字列変換等)をインターフェースの統一されたメソッドとして持つ。

【００１１】(2)データ入出力オブジェクト

磁気ディスク装置、プリンタ等のデータ入出力装置を介して、データ項目オブジェクトとの間でデータ入出力を行うオブジェクト。装置の種別や、装置で用いられるデータ書式等によって、それぞれ異なる種類のデータ入出力オブジェクトが存在する。データ取得およびデータ書き込みの手続きをインターフェースの統一されたメソッドとして持つ。

【００１２】(3)データ処理オブジェクト

データ項目オブジェクトのメソッドを定型的な順序で呼び出すことにより、データ項目オブジェクトに対して特定の定型的処理を行うオブジェクト。定型的処理の内容によりそれぞれ異なる種類のデータ処理オブジェクトが存在する。データ取得およびデータ書き込みの手続きをインターフェースの統一されたメソッドとして持つ。

【００１３】データ入出力オブジェクトおよびデータ処理オブジェクトの、データ取得メソッドおよびデータ書き込みメソッドは、同一のインターフェースによって統一されている。従って、これら２つのメソッドのみを使用する限りは、データ入出力オブジェクトおよびデータ処理オブジェクトは、同一の種類のオブジェクトとして扱う事ができる。以後、データ入出力オブジェクトとデータ処理オブジェクトを包括するオブジェクトの種別として、データ受け渡しオブジェクトという用語を定義し、説明を続ける。

【００１４】データ処理オブジェクトは、データ取得用のデータ受け渡しインスタンスと、データ書き込み用のデータ受け渡しインスタンスを、属性として持つ。データ取得メソッド又はデータ書き込みメソッドが呼び出されると、データ取得用のデータ受け渡しインスタンスのデータ取得メソッドを呼び出して、データ項目インスタンスを取得する。次に、取得したデータ項目インスタンスのメソッドを定型的な順序で呼び出し、その結果として得られたデータ項目インスタンスを、データ書き込み用のデータ受け渡しインスタンスの、データ書き込みメソッドに渡す。ここでメソッド呼び出しが行われたデータ受け渡しオブジェクトがデータ処理オブジェクトである場合には、上記と同様の動作が再帰的に繰り返され

る。

【００１５】オブジェクト接続情報は、データ入出力オブジェクトおよびデータ処理オブジェクトをどのように組合わせて完成プログラムを実現するのかを記述する情報である。完成プログラムで使用するオブジェクトをノードとし、オブジェクト間のデータの流れをノード間の接続とするネットワーク状の情報で構成されている。

【００１６】オブジェクト接続情報が入力されたとき、情報に対して次の2つの検査を行う事により、誤りをプログラム合成前に発見する。

【００１７】(1)情報のネットワーク構造の中の、データ処理オブジェクトのみからなる部分ネットワークに接続されているデータ入出力オブジェクト群の中で、互いに同一のリソースにアクセスするデータ入出力オブジェクトの組が存在するような接続がないかを検査する。このような接続は不正である。

【００１８】(2)上記部分ネットワークの中に、ループ状の接続が存在しないかを検査する。　このような接続は不正である。

【００１９】また、オブジェクト接続情報をユーザが作成する際、情報をグラフィカル・ユーザ・インターフェース等により、対話的に入力する。このとき、ユーザがオブジェクト間のデータの流れを１つ定義する都度、上記検査を行う。

【００２０】検査の結果、正しいと判定されたオブジェクト接続情報からは、コードを生成する。まず、情報に記述されている全てのオブジェクトを生成・初期化する、オブジェクト初期化ルーチンを生成する。次に、情報のネットワーク構造を解析し、ネットワークを、データ処理オブジェクトのみからなる部分ネットワーク群に分解する。完成プログラムでは、各々の部分ネットワークを１つの処理単位として、オブジェクトの制御を行う。各々の部分ネットワークについて、次の４つの動作を行う、部分ネット制御ルーチンが生成される。

【００２１】(1)データ入出力オブジェクトをデータ入力又は出力用にオープンする。

(2)該部分ネットワークに所属する各々のデータ処理オブジェクトに対し、データ入力先とデータ出力先となるデータ受け渡しオブジェクトを渡し、オブジェクト間の接続関係を確立する。

【００２２】(3)データ処理オブジェクトに対し、データ処理依頼を行う。

(4)データ処理が終了したら、データ入出力オブジェクトをクローズする。

【００２３】最後に、データ項目オブジェクト群、データ入出力オブジェクト群、データ処理オブジェクト群、オブジェクト初期化ルーチン、部分ネット制御ルーチン群をコンパイル／リンクすることにより、完成プログラムを得る。

【００２４】

【作用】完成プログラムで処理されるデータをオブジェクトとして抽象化し、さらにメソッドのインターフェースを統一する事で、オブジェクトを受け渡しする、データ入出力オブジェクトおよびデータ処理オブジェクトのメソッドインターフェースが、受け渡しされるデータの形式に非依存となり、メソッドインターフェースを統一する事が可能となる。さらに、データ入出力オブジェクトとデータ処理オブジェクトのメソッドインターフェースを互いに統一することにより、これら２種類のオブジェクトをつなぎ合わせる際に、インターフェースに起因する接続の制限がなくなる。従って、発明が解決しようとする課題の項に示した、第１の課題が解決される。

【００２５】次に、データ処理オブジェクトを複数組み合わせて処理を行う際、これらのデータ処理オブジェクトを交互に動作させる必要があるが、その動作制御をオブジェクト外部から行うのではなく、オブジェクト内部からの再帰的動作で行わせる。オブジェクト外部から行うオブジェクト制御は、オブジェクト間の接続関係の指定と、処理依頼のみであり、単純で分かりやすい制御内容となる。その結果、プログラム合成系が単純で済む上に、合成されたプログラムの保守性も向上する。従って、発明が解決しようとする課題の項に示した、第２の課題が解決される。

【００２６】また、オブジェクト接続情報を、プログラム合成前に検査することにより、オブジェクトの組み合わせの誤りを、プログラムの合成、テストを行う前に知る事ができる。さらに、オブジェクト接続情報をの入力を対話的に行う場合には、ユーザがオブジェクト間のデータの流れを指定する都度、その指定が正当であるか否かが検査される。この結果、ユーザはオブジェクトの組み合わせを定義する作業を行いながら、その場でオブジェクトの組み合わせの誤りを知る事ができる。従って、発明が解決しようとする課題の項に示した、第３の課題が解決される。

【００２７】

【実施例】本発明では、オブジェクト指向プログラミングに関連する用語を次のように定義している。

【００２８】(1)インスタンス

０ないし複数のデータを保持するデータ領域と、手続きを一意に決定するための０ないし複数の、データ値(手続きを実行するサブルーチンが置かれる、計算機上のアドレス等)を保持するデータ領域から成るデータ構造体。

【００２９】(2)メソッド

インスタンスに記述されているデータ値から、一意に決定される手続き。

【００３０】(3)クラス

インスタンスを作成・初期化する手続きと、メソッドから成るプログラムモジュール。

【００３１】(4)オブジェクト

インスタンスおよびクラス。

【００３２】(5)メソッドインターフェース
インスタンスからメソッドを特定しメソッドを呼び出す際の手続き。

【００３３】以下、次のような応用プログラムを合成する場合を例にとり、本発明の実施例を説明する。

【００３４】(1)総務部の人事情報を記録する総務部人事マスタファイルが存在する。このファイルは、従業員レコードの並びによって、従業員の情報が記録されている。また、従業員レコードは、従業員番号をキーにして、あらかじめソートされている。

【００３５】(2)従業員レコードは、従業員番号、従業員名、従業員役職コードの、３つのフィールドから成る。

【００３６】(3)経理部の人事情報を記録する経理部人事マスタファイルが存在する。このファイルの書式は、総務部人事マスタファイルと同一である。

【００３７】(4)プログラムを実行すると、総務部人事マスタファイルと経理部人事マスタファイルの中の、役職コードが001(研修員)である従業員を、研修員一覧票として、帳票出力する。その際出力される従業員は、従業員番号をキーとしてソートされている。

【００３８】図１は、本発明のプログラム合成方法の全体構成図である。図２は本発明のプログラム合成方法で使用するハードウェアの構成図である。図３は、オブジェクト初期化ルーチン生成処理１０５の処理内容を示すフローチャート、図４は、部分接続情報抽出処理１０７の処理内容を示すフローチャート、図５～図８は、部分接続制御ルーチン生成処理１０９の処理内容を示すフローチャートである。

【００３９】以下では、本発明の実施例として、プログラミング言語C++プログラムの合成系を用いて説明するが、必ずしもC++プログラムの合成系に限定されるべきものではなく、あらゆるプログラミング言語のプログラム＊の合成原理を含むものである。ただし、本発明は、オブジェクト指向プログラミング言語のプログラム合成に適用した場合に、効果が最も良く発揮される。

【００４０】本実施例では、データ項目クラス101、データ入出力クラス102、データ処理クラス103は、クラスライブラリに格納されている。ただし、クラスライブラリに格納されているのは、オブジェクトのインターフェースのみを規定する抽象クラスと、多くの応用プログラムで共通的に見られるクラスのみである。クラスライブラリに存在しないクラスは、ユーザが入力する仕様書の情報から、クラスライブラリに格納されているクラスのサブクラスとして生成する。

【００４１】図９に、実施例で用いるクラスライブラリのクラス階層図を示す。DataItem901は、データ項目クラス101のインターフェースを規定する抽象クラスである。メソッド一覧を表1に示す。DataNode905は、データ入出力クラス102およびデータ処理クラス103の共通部分のインターフェースを規定する抽象クラスである。メソッド一覧を表2に示す。DataIO906、DataProcessor907は、それぞれ、データ入出力クラス102と、データ処理クラス103のインターフェースを規定する抽象クラスである。メソッド一覧を表3、表4に示す。その他に、多くの応用プログラムで再利用可能な、KanjiName903、Code Number904等のデータ項目クラス、Merger910、Selector911等のデータ処理クラスはクラスライブラリに格納されており、そのまま応用プログラムの部品として利用する。また、Record902、File908、Sheet909等は、各々データ書式に依存する情報が未定義であり、そのままでは部品として利用出来ないが、プログラム仕様書から入力されるデータ書式情報から、該クラスのサブクラスを生成し、応用プログラムの部品として利用する。

【００４２】

【表1】

DataItem901のメソッド一覧

メソッド名	機能
Compare	データ項目どうしを大小比較する。
SetVal	与えられた文字列をデータ項目値に変換し、セットする。
GetVal	データ項目値を文字列に変換し、返す。
CreInstance	自分自身のコピーを生成する。
GetStrLen	文字列変換時の文字数を取得する。

【００４３】

【表2】

DataNode907のメソッド一覧

メソッド名	機能
GetData	DataItemを取得する。
SetData	DataItemを書き込む。

【００４４】　　＊　＊【表3】

DataIO908のメソッド一覧

メソッド名	機能
GetData	DataItem901を取得する(継承メソッド)。
SetData	DataItem901を書き込む(継承メソッド)。
OpenForRead	読み込み用にオープンする。
OpenForWrite	書き込み用にオープンする。
Close	クローズする。

【００４５】　　※　※【表4】

DataProcessor909のメソッド一覧

メソッド名	機能
Open	入力用および出力用のDataNode907を登録する。
GetData	DataItem901を取得する(継承メソッド)。
SetData	DataItem901を書き込む(継承メソッド)。
Run	入力側DataNodeからのデータが尽きるまでデータ処理を行う。

【００４６】まず、図１０に示すレコード仕様書画面1001、ファイル書式仕様書画面1002、帳票書式仕様書画面1003によってレコード、ファイル、帳票の書式を定義する。レコード仕様書1001は、レコードクラス名、各フィールドの名前、使用するデータ項目の種別(クラス名)等を入力する。本仕様書の情報からは、Record902のサブクラスが生成される。ファイル書式仕様書1002は、ファイルクラス名、使用データ項目の種別等を入力する。本仕様書の情報からは、File908のサブクラスが生成される。帳票書式仕様書1003は、帳票クラス名、使用データ項目の種別、題名文字列(1枚目の帳票の先頭行に印刷される文字列)、ヘッダ文字列(帳票の各ページ先頭行に印刷される文字列)、フッタ文字列（帳票の各ページ末尾行に印刷される文字列)等を入力する。本仕様書の情報からは、Sheet909のサブクラスが生成される。

【００４７】次に、図１１に示すオブジェクト接続情報入力画面によって、オブジェクト接続情報104を作成する。入力画面では、オブジェクトの組み合わせと、個々のオブジェクトの詳細情報を入力する。まず、1101に示すネットワーク図により、完成プログラムにおけるオブジェクト間のデータの流れを定義する。1102～1106に示すアイコンは、完成プログラムで使用されるオブジェクトを表現しており、１つのアイコンが一つのインスタンスに対応する。アイコン間を結ぶ矢印は、インスタンス間のデータの流れを表現する。以後、ネットワーク図をシステムフロー図と呼ぶ。システムフロー図1101では、次のようなデータの流れが定義されている。

【００４８】(1)総務部人事マスタファイル1102は、マージ部品1104からの要求を受けるとファイルからデータを読みだし、結果として得られたレコードインスタンスを返す。

【００４９】(2)経理部人事マスタファイル1103は、マージ部品1104からの要求を受けるとファイルからデータを読みだし、結果として得られたレコードインスタンスを返す。

【００５０】(3)マージ部品1104は、総務部人事マスタファイル1102と、経理部人事マスタファイル1103からレコードインスタンスを取得し、該レコードインスタンスの大小比較メソッドを呼び出す。大小比較した結果、より(キー項目が)小さいと判断されたレコードインスタンスを出力する。出力したレコードインスタンスが総務部人事マスタファイル1102のものならば、総務部人事マスタファイル1102から1レコード取得し、経理部人事マスタファイル1103のものならば、経理部人事マスタファイ

ル1103から1レコード取得する。該操作を繰り返すことにより総務部人事マスタファイル1102、経理部人事マスタファイル1103双方のレコードが、キー項目がソートされた状態で順次出力される。

【００５１】(4)選択部品1105は、マージ部品から入力したレコードの特定フィールドが特定の値である場合のみ得られたレコードを研修員一覧票1106に出力する。(5)研修員一覧票1106は、渡されたレコードを帳票に出力する。

【００５２】次に、1107～1111に示す表により、個々のプログラム部品インスタンスの詳細定義を行う。全ての詳細定義1102から1111において、個々のインスタンスの所属クラスおよびインスタンス名を定義するが、その他の付加情報を定義しなければならないインスタンスも存在する。総務人事マスタファイル1102及び、経理部人事マスタファイル1103の詳細定義1107、1108では、使用するファイルのファイル名を定義する。選択部品1105の詳細定義1110では、レコードの中の、最初から何番目のフィールドを検査するのかを定義する検査フィールド番号と、監査したフィールドがどのような値である場合にレコードを出力するかを定義する適合値の、２つの付加情報を定義する。

【００５３】検査の結果、オブジェクト接続情報104には、正当な接続情報が格納される。オブジェクト接続情報104からは、部品初期化ルーチン106と、部分接続制御ルーチン群110が生成される。本実施例では、部品初期化ルーチン106と、部分接続制御ルーチン群110は、一体化されたメインルーチンとして生成される。生成されるメインルーチンを図１２に示す。

【００５４】まず、図３のフローチャートに基づき、図１１において記述された全てのインスタンスを生成・初期化する1201が生成される。まず、３０１でオブジェクト接続情報に未処理のオブジェクトが残っているかを調べ、オブジェクト接続情報内の全てのオブジェクトを処理し終えたらオブジェクト初期化ルーチン生成処理１０５を終了する。未処理のオブジェクトが残っていれば、３０２で上記未処理オブジェクトをオブジェクト接続情報内から取得し、３０３で取得したオブジェクトの初期化ルーチン呼出しコードを出力する。出力が終わると、３０１に戻る。

【００５５】次に、図４のフローチャートに基づき、データ処理オブジェクトのみからなる部分接続と、部分接続に直接接続されているデータ入出力オブジェクトのみからなる部分接続情報が抽出される。まず、４０１で、オブジェクト接続情報からデータ処理オブジェクトのみからなる部分を抽出する。４０２で、上記部分が抽出できたかを判定する。抽出されなかったと判断された場合は、部分接続情報抽出処理１０７を終了し、抽出に成功したと判定され時は４０３に進む。４０３では４０１で抽出した、データ処理オブジェクトのみからなる部分と、上記部分に直接接続されているデータ入出力オブジェクトから成る部分を併せた部分接続情報を出力する。出力が終了すると、４０１に戻る。

【００５６】本実施例の図１１からは、マージオブジェクト1104と、選択オブジェクト1105からなる部分が抽出され、部分に直接接続されているデータ入出力オブジェクトも含めた、総務部人事ファイル1102、経理部人事マスタファイル1103、マージオブジェクト1104、選択オブジェクト1105、研修員一覧票1106からなる部分接続情報が出力される。

【００５７】上記部分接続情報から、図５のフローチャートに基づき、部分接続情報に含まれるオブジェクトを制御する部分1202が生成される。まず、５０１で図４のフローチャートに基づいて抽出された部分接続情報がまだ残っているかを判定する。残っていなければ部分接続制御ルーチン生成処理１０８を終了し、残っているならば、５０２に進む。５０２では上記部分接続情報を取得し、部分接続初期化コード生成処理５０３、データ処理依頼コード生成処理５０４、リソース解放コード生成処理５０５を順次行なった上で５０１に戻る。

【００５８】図６は、部分接続初期化コード生成処理５０３の詳細を示すフローチャートである。まず、６０１で、５０２において取得した部分接続情報に未処理のオブジェクトが残っているかを判定する。残っていないならば部分接続初期化コード生成処理５０３を終了し、残っているならば６０２に進む。６０２でオブジェクト種別を判別し、次に６０３でそれがデータ入出力オブジェクトかを判定する。データ入出力オブジェクトでないならば６０４に進み、データ入出力オブジェクトならば６０５に進む。６０４ではデータ処理オブジェクトのオブジェクト接続オブジェクトを呼び出すコードを出力し、６０１に戻る。６０５ではデータ入出力オブジェクトのオープンメソッドを呼び出すコードを出力し、６０１に戻る。上記フローチャートにより、図１２に示したメインルーチンの例では１２０３の部分が生成される。

【００５９】図７は、データ処理依頼コード生成処理５０４の詳細を示すフローチャートである。まず７０１で、５０２において取得した部分接続情報から任意のデータ処理オブジェクトを抽出する。次に７０２で上記データ処理オブジェクトのデータ処理メソッドを呼び出すコードを出力する。上記フローチャートにより、図１２に示したメインルーチンの例では１２０４の部分が生成される。

【００６０】図８は、リソース解放コード生成処理５０５の詳細を示すフローチャートである。まず８０１で、５０２において取得した部分接続情報に未処理のデータ入出力オブジェクトが残っているかを判定する。データ入出力オブジェクトが残っていないならばリソース解放コード生成処理５０５を終了し、残っているならば８０２に進む。８０２でオブジェクト種別を判別し、次に８

０３でデータ入出力オブジェクトのクローズメソッドを呼び出すコードを出力する。上記フローチャートにより、図１２に示したメインルーチンの例では１２０５が生成される。

【００６１】図１における部品初期化ルーチン１０６は図１２の１２０１に、図１における部分接続制御ルーチン１１０は図１２の１２０２に対応する。本実施例では、部品初期化ルーチン１０６及び部分接続制御ルーチン１１０は１つのメインルーチンとして生成される。最後に、本実施例で生成したサブクラス及びメインルーチンをコンパイルし、さらに最初に説明したクラスライブラリをリンクすることにより、完成プログラムを得る。

【００６２】次に、第２の実施例を説明する。本実施例では、複数のデータ処理オブジェクトが、同時にシステム上の同一のリソースにアクセスしないように、オブジェクト接続情報を検査する。図13に示す接続例では、データ処理オブジェクト1301と、データ処理オブジェクト1302が、帳票1303と、帳票1304に処理結果を出力している。完成プログラムでは、データ処理オブジェクト1301とデータ処理オブジェクト1302は同時に動作する事になるため、帳票1303と帳票1304が同一のプリンタから出力されると、正常な結果を得ることができない。

【００６３】このような、不正な接続を検出するためには、システムフロー図を、データ処理オブジェクトのみからなる部分ネットワークに分解し、同一の該部分ネットワークに接続されているデータ入出力オブジェクトで、同一リソースを使用しているものがないかを検査する。

【００６４】次に、第３の実施例を説明する。発明では、完成プログラムにおいて、処理途中にあるデータが失われることのないように、オブジェクト接続情報を検査する。図１４に示す接続例では、データ処理オブジェクト1403がデータ処理オブジェクト1401から2回続けてデータを取得すると、データ処理オブジェクト1401からデータ処理オブジェクト1402へ、2回続けてデータが出力される可能性がある。データ処理オブジェクト1402の処理結果データがデータ処理オブジェクト1403に渡される前に、次のデータが送られてきてしまうため、処理途中のデータが次のデータ処理オブジェクトに渡される前に消滅してしまう。

【００６５】このような、不正な接続を検出するためには、システムフロー図を、データ処理オブジェクトのみからなる部分ネットワークに分解し、該部分ネットワークにループ接続がないかを検査する。

【００６６】次に、第４の実施例を説明する。該発明では、システムフロー図1101をユーザと対話的に入力する際、ユーザがオブジェクト間のデータの流れを定義する都度、検査を行う。検査が必要となるのは、ユーザが次の動作を行った時である。

【００６７】(1)オブジェクトを示すアイコン間にデータの流れを示すアークを引いた時。
(2)アークで接続されているアイコンの種類又は名称を変更した時。
(3)アイコン間のアークの引き方を変更した時。

【００６８】上記以外の、オブジェクト間のデータの流れが変更されない操作をユーザが行った場合は、検査を行う必要がない。検査が行われる操作を、あらかじめ限定しておくことにより、システムフロー図を入力するエディタの性能を著しく損なうことなく、かつ、不正なオブジェクトの接続を、システムフロー図の作成時に、その場で知る事ができる。

【００６９】

【発明の効果】本発明によれば、ユーザがプログラム部品を組み合わせることによってプログラムを自動合成する際、部品の組み合わせを意識することなく容易に組み合わせ定義が行え、かつ、部品間のデータ受け渡しにテンポラリファイルを介さない、高性能で、運用の手間がかからないプログラムを自動合成する事ができる。

【図面の簡単な説明】

【図１】プログラム合成方法の、処理手順の全体図。

【図２】ハードウェア構成図。

【図３】図１に示すオブジェクト初期化ルーチン生成処理105のフローチャート。

【図４】図１に示す部分接続情報抽出処理107のフローチャート。

【図５】図１に示す部分接続制御ルーチン生成処理109のフローチャート。

【図６】図５に示す部分接続初期化コード生成処理501のフローチャート。

【図７】図５に示すデータ処理依頼コード生成処理502のフローチャート。

【図８】図５に示すリソース開放コード生成処理503のフローチャート。

【図９】実施例のクラスライブラリのクラス階層図。

【図１０】実施例のプログラム仕様書入力画面。

【図１１】実施例のオブジェクト接続情報入力画面。

【図１２】実施例で生成されるメインルーチン。

【図１３】同一リソースへの同時アクセスが行われる、不正な接続例。

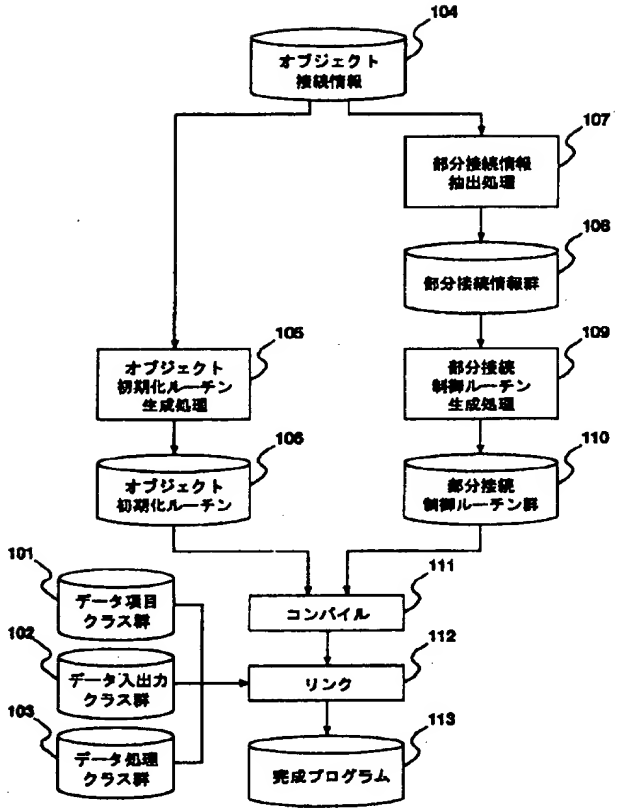
【図１４】処理途中データが失われる、不正な接続例。

【符号の説明】

101…データ項目クラス群、102…データ入出力クラス群、103…データ処理クラス群、104…オブジェクト接続情報、105…オブジェクト初期化ルーチン生成処理、106…オブジェクト初期化ルーチン、107…部分接続情報抽出処理、108…部分接続情報群、109…部分接続制御ルーチン生成処理、110…部分接続制御ルーチン群、111…コンパイル、112…リンク、113…完成プログラム

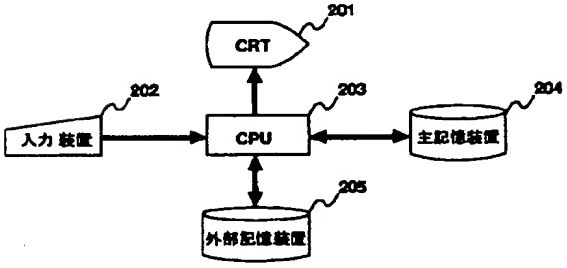
【図１】

図1



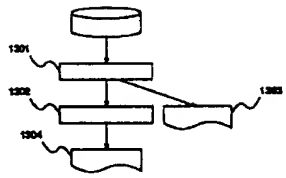
【図２】

図２



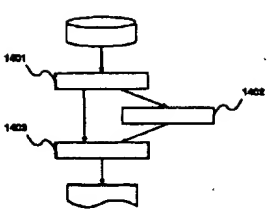
【図１３】

図１３



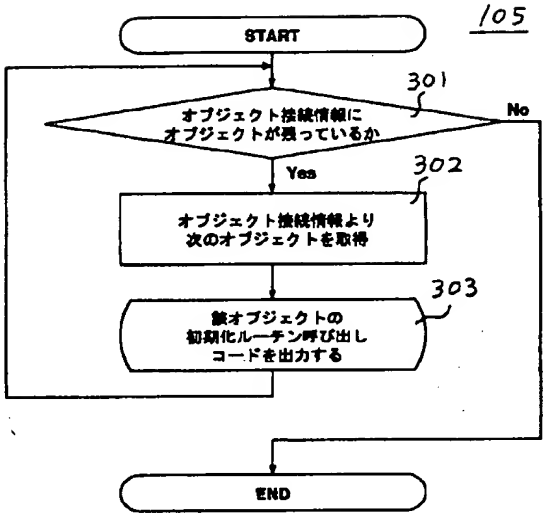
【図１４】

図１４



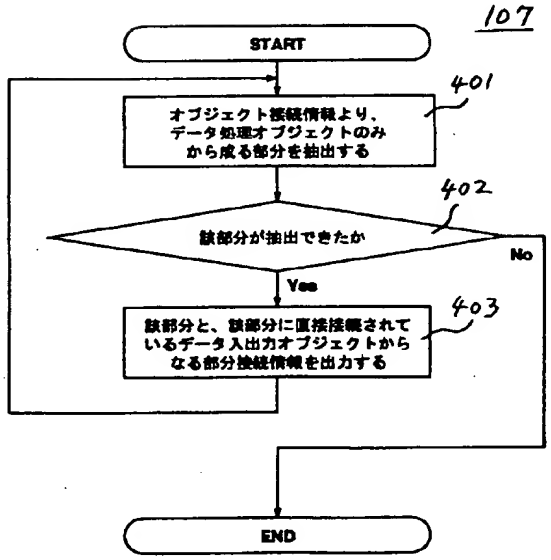
【図3】

図3


105
START
301
オブジェクト接続情報に
オブジェクトが残っているか
No
Yes
302
オブジェクト接続情報より
次のオブジェクトを取得
303
該オブジェクトの
初期化ルーチン呼び出し
コードを出力する
END

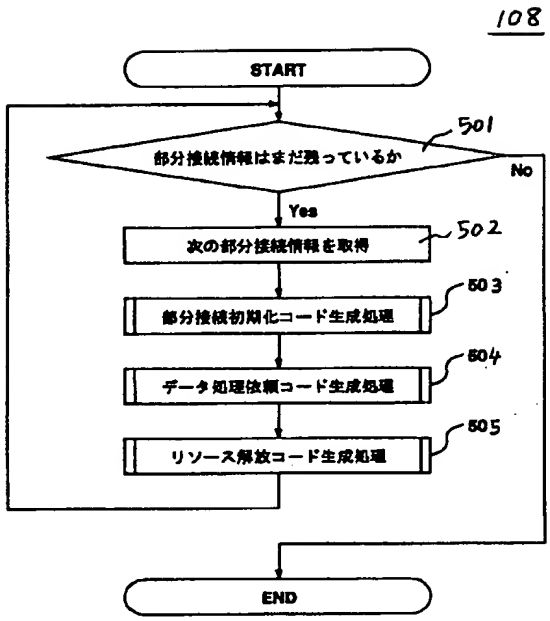
【図4】

図4


107
START
401
オブジェクト接続情報より、
データ処理オブジェクトのみ
から成る部分を抽出する
402
該部分が抽出できたか
No
Yes
403
該部分と、該部分に直接接続されて
いるデータ入出力オブジェクトから
なる部分接続情報を出力する
END

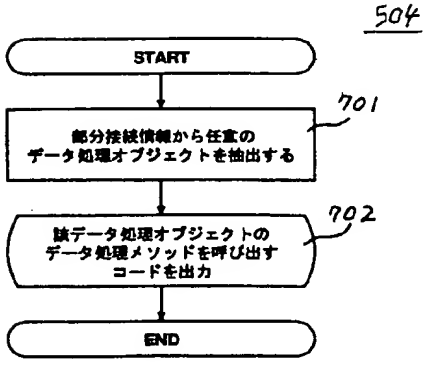
【図5】

図5


108
START
501
部分接続情報はまだ残っているか
No
Yes
502
次の部分接続情報を取得
503
部分接続初期化コード生成処理
504
データ処理依頼コード生成処理
505
リソース解放コード生成処理
END

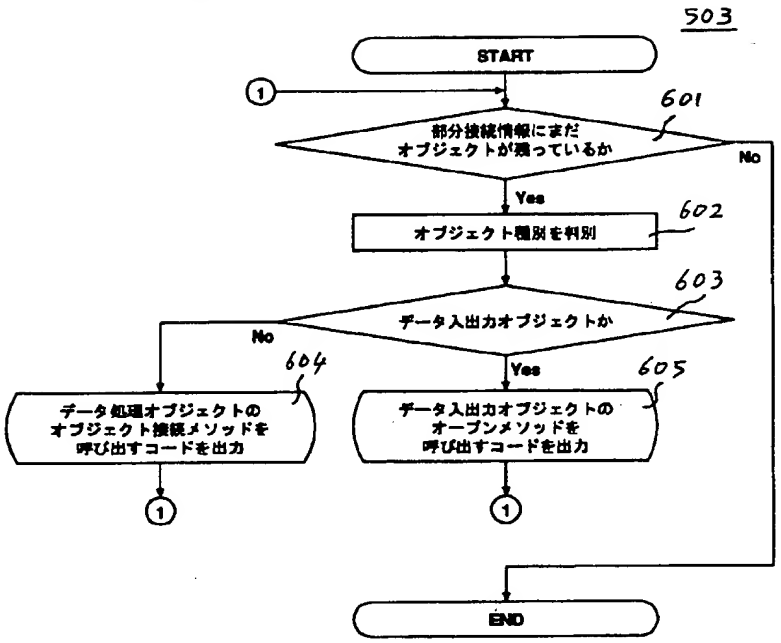
【図7】

図7


504
START
701
部分接続情報から任意の
データ処理オブジェクトを抽出する
702
該データ処理オブジェクトの
データ処理メソッドを呼び出す
コードを出力
END

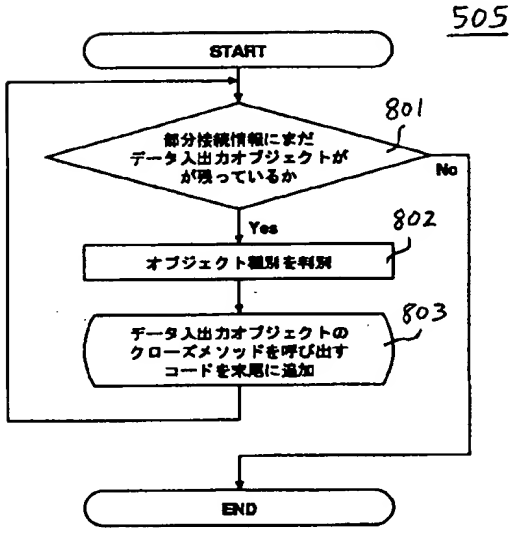
【図6】

図6


503
START
1
601
部分接続情報にまだ
オブジェクトが残っているか
No
Yes
602
オブジェクト種別を判別
603
データ入出力オブジェクトか
No
Yes
604
データ処理オブジェクトの
オブジェクト接続メソッドを
呼び出すコードを出力
605
データ入出力オブジェクトの
オープンメソッドを
呼び出すコードを出力
1
1
END

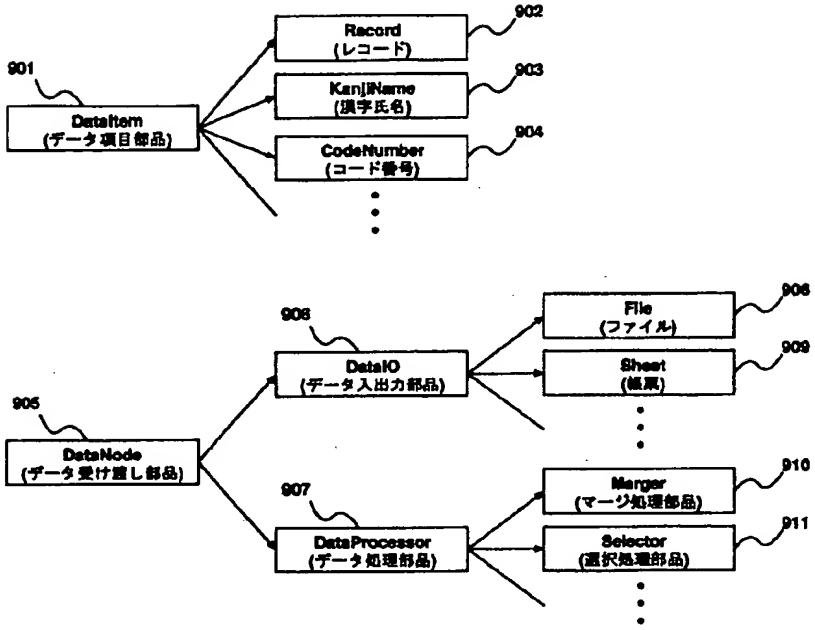
【図8】

図8


505
START
801
部分接続情報にまだ
データ入出力オブジェクトが
が残っているか
No
Yes
802
オブジェクト種別を判別
803
データ入出力オブジェクトの
クローズメソッドを呼び出す
コードを末尾に追加
END

【図9】

図9



【図12】

図12

```
void main(void)
{
    JinjiFile     schemaJinjiFile;
    JinjiFile     keiriJinjiFile;
    Merger        merger1;
    Selector      selector1;
    JinjiSheet    hensshuinList;

    schemaJinjiFile.OpenForRead("schemajinji.dat");
    keiriJinjiFile.OpenForRead("keirijinji.dat");
    merger1       .Open(schemaJinjiFile,keiriJinjiFile,selector1);
    selector1     .Open(merger1,hensshuinList);
    hensshuinList .OpenForWrite();

    selector1->Run();

    schemaJinjiFile.Close();
    keiriJinjiFile.Close();
    hensshuinList .Close();
}
```

【図１０】

図１０

1001

レコード書式仕様書

レコードクラス名	Jugyoin
備考	従業員レコード

レコード書式

キー項目	フィールド名	データ項目種別	備考
●	code	CodeNumber	従業員番号
	name	KanjiName	従業員名
	classCode	CodeNumber	役職コード

1002

ファイル書式仕様書

ファイルクラス名	JinjiFile
備考	人事ファイル
使用データ項目	Jugyoin

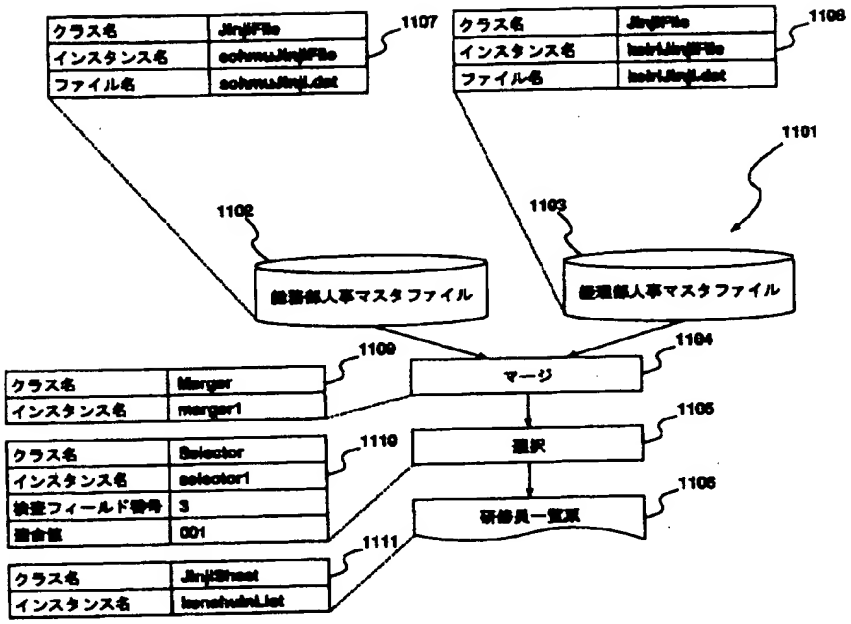
1003

帳票書式仕様書

ファイルクラス名	JinjiSheet
備考	人事データ一覧表
使用データ項目	Jugyoin
題名文字列	＊＊＊ 人事データ一覧 ＊＊＊
ヘッダ文字列	従業員番号　　従業員名　　役職コード
フッタ文字列	

【図１１】

図１１


クラス名
インスタンス名
ファイル名
1107
1108
1101
1102
1103
1109
1104
マージ
1110
1105
選択
1106
1111